technische universität
dortmund

fakultät für
informatik

computer
science 12

christian.hakert [☺] tu-dortmund.de
marcel.ebbrecht [☺] tu-dortmund.de

Exercise Sheet 4

Discussion starts from Monday, May 25, 2020

4.1   Classes of Multiprocessor Systems

Fill the classification of multiprocessor systems according to Flynn into the given table and describe one important characteristic of each class.

		instruction streams	
		1	>1
data streams	1		
	>1		

Class: _____
Characteristic:

Class: _____
Characteristic:

Class: _____
Characteristic:

Class: _____
Characteristic:

4.2 Cache Coherence

(a) Give a brief definition of *cache coherence.*

(b) Name the two basic strategies provided by cache coherence protocols to maintain coherence in a system. You are not expected to explain these strategies.

(c) Cache coherence for write-back caches can be achieved by means of the MSI protocol introduced in the lecture. Hereafter, a state diagram of a cache block is depicted under the MSI protocol. Add the missing labels for states as well as for state transitions. The input/output alphabet of the state machine is given in the following table.

inputs	outputs
CPU read miss	Read Miss
CPU read hit	Write Miss
CPU write miss	Data Write Back
CPU write hit	
Invalidate	
Fetch	
Fetch/Invalidate	

invalid

CPU write miss
write miss

CPU read miss
read miss

CPU write / read hit

CPU write miss
_____ / **write miss**

data write back / **read miss**
Fetch
data write back

CPU read hit

read miss
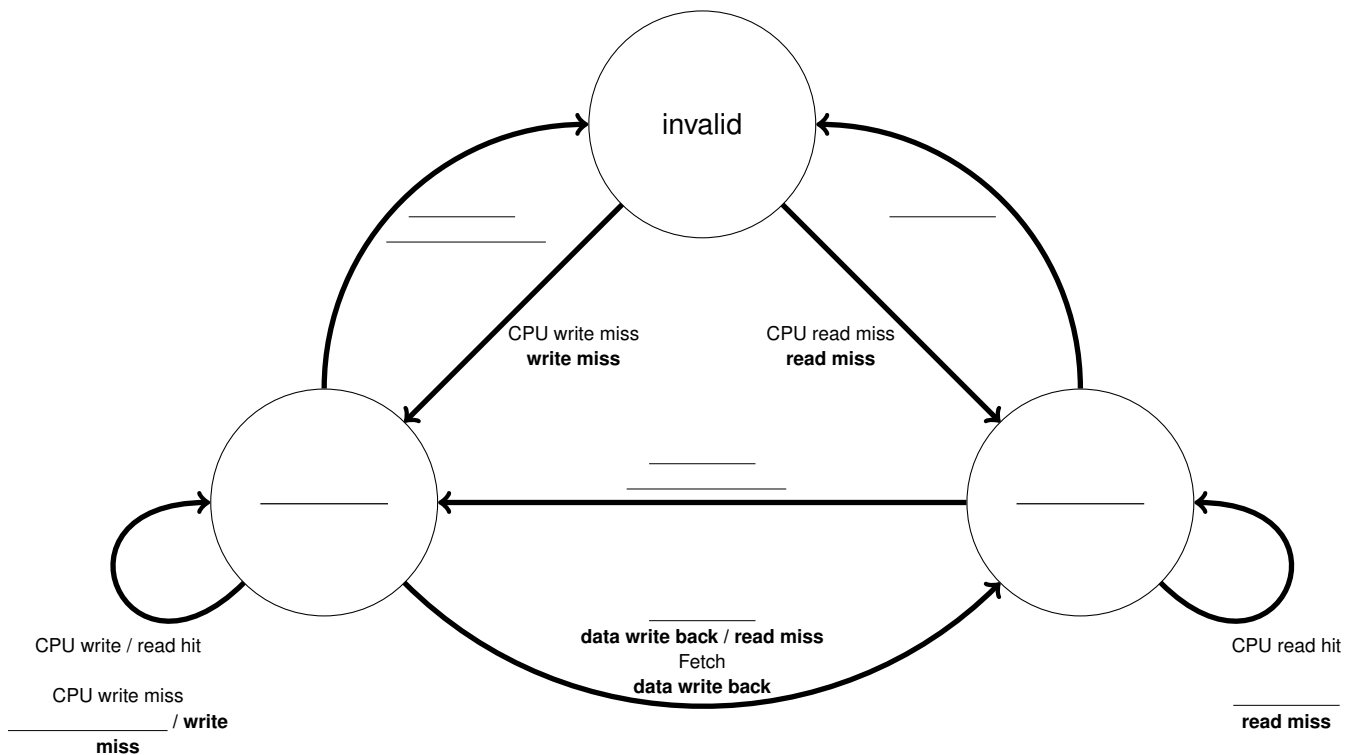
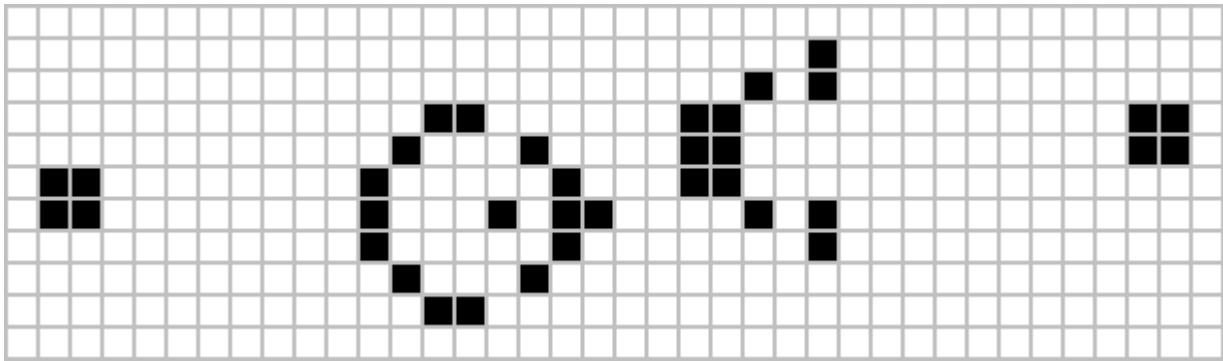## 4.3 Parallelization of Conway's Game of Life with OpenMP

Abbildung 1: Game of Life Example - Gun Slider Pattern [www.wikipedia.org]

The grid consists of $v \times h$ elements, as shown in Figure 1. Each cell can be either alive (value 1, black field) or dead (value 0, white field). Starting from an initial configuration, the grid is iteratively updated. After the update, the current iteration is finished and the update process is repeated until a certain threshold is reached. For the update process, certain rules are given:

- A dead cell with exactly three living neighbors is reborn in the next iteration.

- Living cells with less than two living neighbors die by solitude in the next iteration .

- A living cell with two or three living neighbors stays alive in the next iteration.

- Living cells with more than three living neighbors die by overpopulation in the next iteration.

Analyze the source code of the Game of Life provided in the SVN repository ("04_2.c" or "04_2.cpp").
Optional assignment: Implement your own version of the Game of Life at home using C or C++.

a. Ensure that each parallelized program is semantically correct and leads to the same result as a sequential version.

b. Execute the examples RA and BIG in the file "muster.txt" located in the SVN repository for different numbers of parallel threads.